Execution Monitoring and Replanning with Incremental and Collaborative Scheduling

David E. Wilkins[1], Stephen F. Smith[2], Laurence A. Kramer[2], Thomas J. Lee[1] and Timothy W. Rauenbusch[1]

[1]Artificial Intelligence Center
SRI International
Menlo Park, CA 94025
wilkins,tomlee,rauenbusch@ai.sri.com

[2]The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
sfs,lkramer@cs.cmu.edu

Abstract

We describe the Flight Manager Assistant (FMA), a prototype system, designed to support real-time management of airlift operations at the USAF Air Mobility Command (AMC). In current practice, AMC flight managers are assigned to manage individual air missions. They tend to be overburdened with associated data monitoring and constraint checking, and generally react to detected problems in a local, myopic fashion. Consequently decisions taken for one mission can often have deleterious effects on others. FMA combines two key capabilities for overcoming these problems: (1) intelligent monitoring of incoming information (for example, weather, airport operations, aircraft status) and recognizing those situations that require corrective action, and (2) dynamic rescheduling of missions in response to detected problems, both to understand the global implications of changed circumstances and to determine appropriate rescheduling actions. FMA builds on two of our existing technologies: an execution-monitoring framework previously applied to small-unit operations and control of robots, and a dynamic scheduling tool that is transitioning into operational use in AMC's Tanker/Airlift Control Center. FMA's dynamic mediation module provides for collaborative mission management by different planning and execution offices by structuring communication for decision making.

Introduction and Problem Statement

Management of flight operations at the United States Air Force Air Mobility Command (AMC) is a challenging problem. AMC typically flies several thousand missions worldwide on a weekly basis (more in a crisis situation), involving several hundreds of aircraft and comparable numbers of aircrews. The execution of any given mission requires attention to a broad range of constraints relating to the mission's requirements (e.g., delivery dates, cargo type and weight), resource availability (e.g., aircraft, aircrews, airports, diplomatic clearances) and usage constraints (e.g., crew duty day restrictions and scheduled return dates, aircraft speed, range, and capacity, airspace restrictions). Although missions are planned and globally scheduled to satisfy such constraints, the dynamics of execution regularly forces changes. Aircraft break down, airports become unavailable due to weather, missions become delayed due to diplomatic clearance problems, etc., and all such events can warrant reassessment of previous allocation decisions. In such execution-driven rescheduling contexts, it is important to weigh potential recovery options against their prospective impact on future operations, and to take actions that continue to make the most effective global use of AMC assets.

In current practice, management of flight operations at AMC is a stove-piped process, where planning and execution are treated as sequential steps and information flows in one direction (from planning to execution). New mission requirements flow into AMC's planning offices on a continuous basis; and as they do aircraft and aircrews are incrementally allocated to support new missions in accordance with associated priorities and as resource availability allows. When a mission gets to within 24 hours of execution, it is "pushed" from the planning side of AMC to the execution office, and becomes the responsibility of an individual flight manager. AMC flight managers take responsibility for checking to insure that all mission constraints remain satisfied before and during execution, and as problems are detected, they diagnose and revise mission plans to facilitate mission continuation and/or recovery. Unfortunately, AMC flight managers are not well supported in this execution management task. Some alerting tools do exist for signaling certain kinds of problems; but there is generally no ability to differentiate routine checks from exceptional events (i.e., everything shows up red), and no ability to detect more complex, compound conditions. Flight managers are typically overburdened by the data monitoring and constraint checking activities that are required to ensure the continuing viability of executing missions. Furthermore, when problematic situations are detected, flight managers have no visibility of the larger AMC operating picture, and must take recovery actions without regard to potential interactions with other missions. As a result, execution management often proceeds in fire-fighting mode, where putting out one fire ignites the next one.

For the past several years, we have been engaged in the development of technologies that we believe can provide a basis for more effective flight management. At Carnegie Mellon University (CMU) we have been developing the

AMC Allocator, a dynamic scheduling tool for day-to-day management of airlift and tanker schedules [Kramer & Smith 2002, Smith et. al 2004]. The AMC Allocator provides a range of capabilities for incrementally revising schedules to accommodate new or changed requirements, with continued emphasis on efficient resource utilization. It is currently transitioning into use as a "planning" tool in the Tanker/Airlift Control center at AMC. At SRI, we have been developing the Small Unit Operations Execution Assistant (SUO-EA), which monitors large volumes of situational data and gets urgent, plan-aware alerts to the right users [Wilkins et al. 2003]. SUO-EA has been successfully demonstrated in both the DARPA SUO program and ONR UCAV program. Also at SRI, we have developed technologies for incremental negotiation and coalition formation technology within the DARPA Autonomous Negotiating Teams program and the ONR UCAV program [Ortiz et al. 2003]. Finally, SRI's Open Agent Architecture (OAA) [Cheyer&Martin 2001] provides a robust integration infra-structure which has been used in dozens of programs and applications.

In this paper, we describe the Flight Manager Assistant (FMA), a system that integrates the above set of technology components to provide a flexible, mixed-initiative tool for real-time flight management. Through a coupling of execution monitoring capabilities with a global dynamic scheduler, the FMA is designed to promote a more integrated, and hence more informed basis for detecting and responding to exceptional execution events. The FMA actively monitors data information sources for expectations it derives from the current schedule, recognizes deviations immediately, and applies policies for responding to deviations. Responses to significant deviations may alert the user to take control. Other options might include automated responses (when permitted by policy), or invoking the scheduler to explore alternative rescheduling options. By integrating status update information with the current schedule, the FMA indicates the important consequences of detected events on current and future operations. Through generation and comparison of alternative schedule repair options (either through interaction with the user or automatically), the FMA supports determination of globally coherent recovery actions while also promoting schedule changes that minimize disruption to other missions whenever possible. A given schedule repair process may also initiate and assist a collaboration between the user responsible for execution and the users who planned the missions. Finally, the FMA can provide automated support for implementing the human-selected response. The FMA continuously reacts to new information while interspersing its proactive pursuit of response procedures.

The broad goal of the FMA project has been to develop technology that enables increased organizational responsiveness and effectiveness in managing the dynamics of mission operations. In our view, there are two key factors to realizing this goal:

- *Increased automation.* Ubiquitous computers, data sources, and reliable, high-bandwidth communication networks are providing too much information for humans to monitor. In our vision, flight managers will rely on an automated execution aid to monitor the large (and ever increasing) volume of incoming information. By understanding the plan and situation, such an execution aid will consider the outputs of multiple monitoring techniques and tools, and then judge when the user should be alerted. Good judgment avoids overalerting. There may be many exceptions noted in the current plan by various AMC monitoring tools – the FMA recognizes which are most important, focuses the human on those, and assists with developing responses.
- *Closing the loop between planning and execution.* The ability to effectively respond to important alerts requires access to the global state of current and planned future operations, and to the rationale that underlies current mission plans/schedules. In our vision, flight managers will utilize dynamic scheduling tools to understand the consequences of detected events, to generate alternative reactions and evaluate the impact of each, and to provide a basis for negotiating mission requirements—the FMA provides these sorts of capabilities and enables a flight manager to apply a more global perspective in determining how to respond. The FMA also alerts originating planners to problems with their missions and provides support for them to contribute information relevant to execution decisions and achieve globally beneficial changes to individual mission plans.

The current FMA prototype is comprised of two principal components: a Flight Manager Executive (built from SRI's SUO-EA system) and a Dynamic Scheduler (derived from CMU's AMC Allocator system). We have demonstrated this prototype on a series of execution management vignettes, using actual (full scale) AMC schedules pulled from AMC's Consolidated Air Mobility Planning System (CAMPS), and representative (but scripted) execution data streams. A third Dynamic Mediation component (based on SRI's incremental negotiation techniques) has undergone preliminary proof of concept testing.

In the sections below we describe these components in more detail, and give an indication of the application's status and potential for transition.

FMA Architecture

The FMA architecture features actors. There is an actor for each participant in the decision-making process. The FMA is configurable for arbitrary sets of decision makers. A typical configuration includes at least one actor for the Execution Office and for each planning office (e.g., SAAM, Channel). Figure 1 depicts the various actors in a common configuration of our Flight Manager Assistant. We designed the software architecture for the various SRI and CMU components, and decided to use OAA to communicate between the various software agents in our architecture. Our system, the Flight Manager Assistant, is composed of four software modules:

- GUI
- Executive (Exception Handler)
- Dynamic Scheduler (DS)
- Dynamic Mediator (DM)

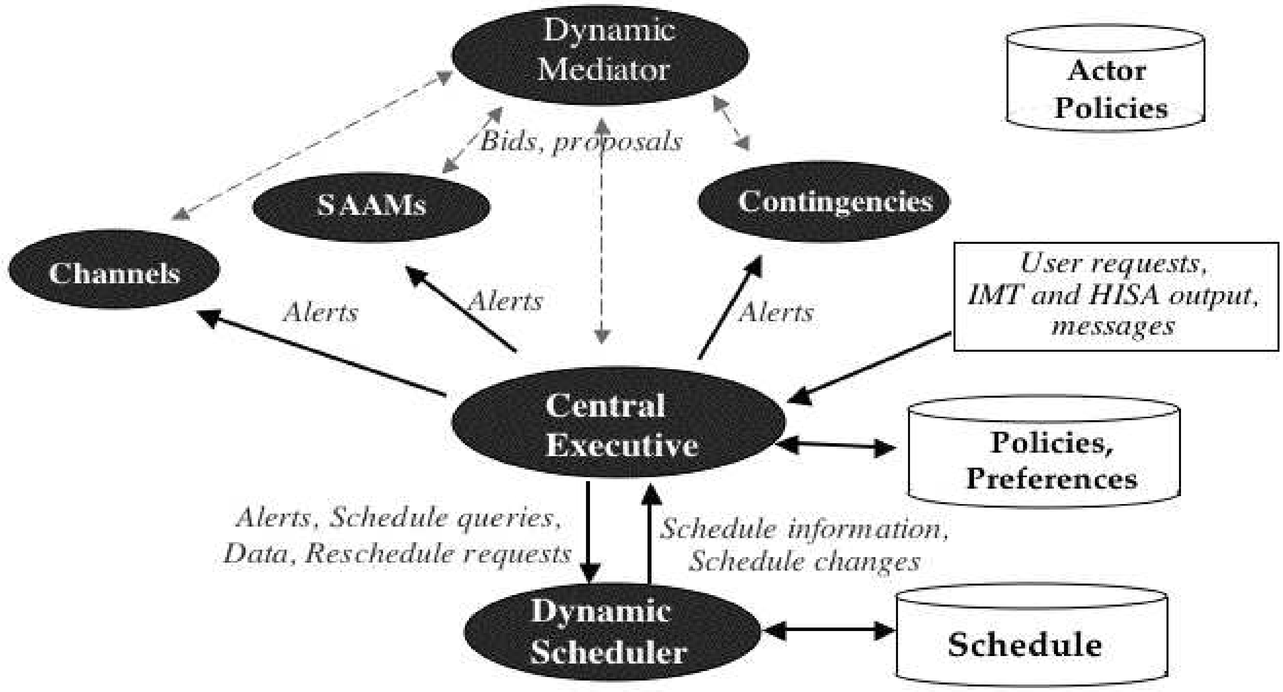

Figure 1: FMA Architecture. Arrows represent message and information flow; every agent communicates with the Actor Policies (arrows omitted). FMA monitors the output of AMC tools IMT and HISA.

The DS is an FMA actor. Each other actor is an instantiation of the Executive, with its own GUI and value-of-information (VOI) functions that determine the alerts received and their priority.

The inputs the FMA monitors come from various AMC tools and messages from other actors and external agents. For example, one tool detects and reports maximum on ground (MOG) conflicts at airbases.

Executive. The key problem for the Executive is that algorithms that alert on constraint violations and threats in a straightforward manner inundate the user in dynamic domains. Unwanted alerts are a problem in many domains, from medicine to transportation to battle command. An execution aid that gives alerts every few seconds will quickly be discarded by the user in stressful situations (if not immediately). To be useful, an execution aid must produce high-value, user-appropriate alerts. Alerts and their presentation may also have to be adjusted to the situation, including the user's cognitive state (or the computational state of a software agent). For example, in high-stress situations, tolerances could be increased or certain types of alerts might be ignored or postponed.

Our approach is grounded in the concept of determining the value of an alert. First, the system must estimate the value of new information to the user. We use the term value of information (VOI) to refer to the pragmatic import the information has relative to its receiver. We assume that the practical value of information derives from its usefulness in making informed decisions. However, alerting the user to all valuable information could have a negative impact in certain situations, such as when the alert distracts the user from more important tasks, or when too many alerts overwhelm the user. We therefore introduce the concept of value of an alert (VOA), which is the pragmatic import (for making informed decisions) of taking an action to focus the user's

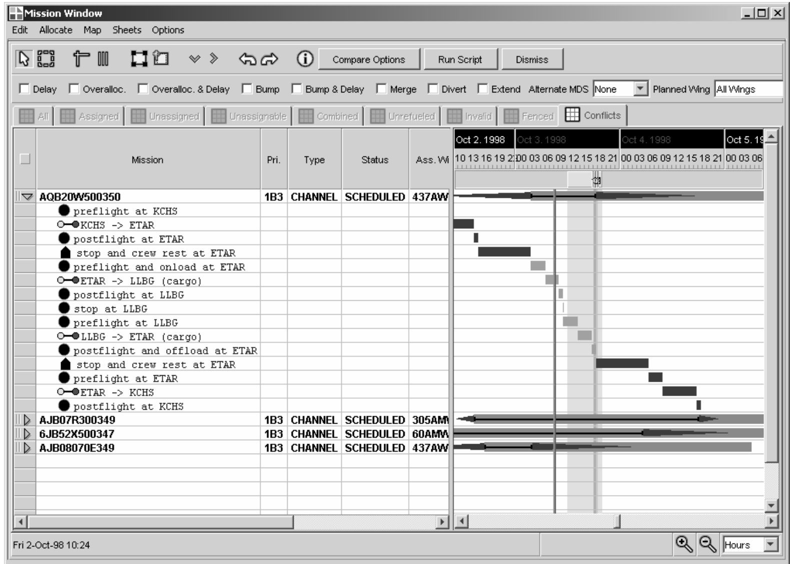

Figure 2: Screenshot of the Dynamic Scheduler

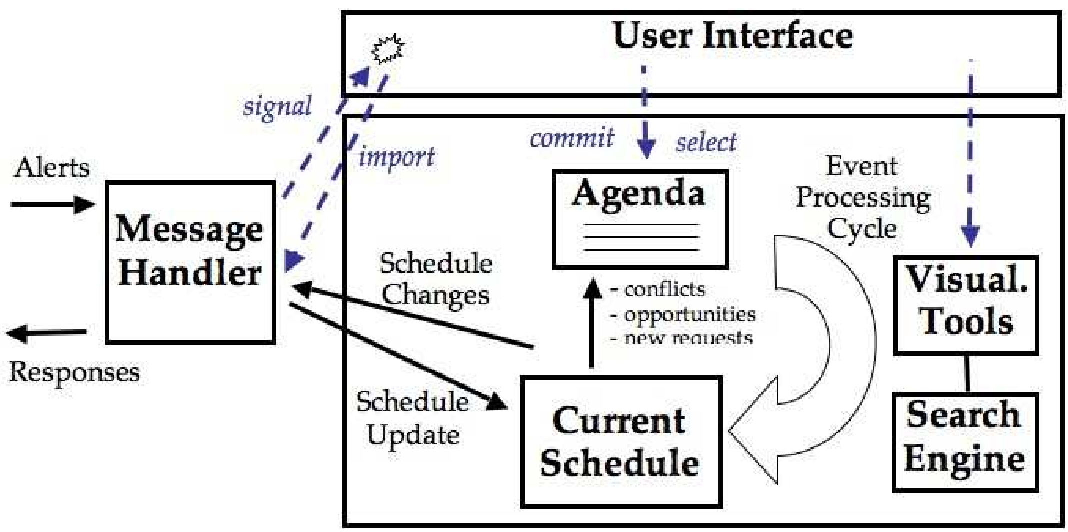

Figure 3: Internal architecture of the FMA Dynamic Scheduler

attention on a piece of information. VOA takes VOI into account but weighs it against the costs and benefits of interrupting the user. If the user is busy doing something significantly more important, then issuing an alert might not be valuable, even when VOI is high.

Our monitoring framework integrates many domain-specific and task-specific monitoring techniques and then uses the concept of value of an alert to avoid operator overload. We have used this framework to implement Execution Assistants (EAs) in three different dynamic, data-rich, real-world domains to assist a human in monitoring team behavior. One domain (Army small unit operations) has hundreds of mobile, geographically distributed agents, a combination of humans, robots, and vehicles. The second domain (teams of unmanned ground and air vehicles) has a handful of cooperating robots. Both domains involve unpredictable adversaries in the vicinity. The application to integrated flight management at AMC represents our third application. Our approach customizes monitoring behavior for each specific task, plan, and situation, as well as for user preferences.

Dynamic Scheduler. The dynamic scheduler (DS) provides capabilities for assessing the broader impact of events that have caused alerts and for determining appropriate mitigating changes to the current airlift schedule. As indicated earlier, the DS extends the technology and software first implemented in the AMC Allocator [Kramer & Smith 2002, Smith et al. 2004], a system for day-to-day management of airlift and tanker schedules that is now embedded as an operational module in the AMC CAMPS mission planning system. At its core, the AMC Allocator utilizes incremental, constraint-based scheduling techniques that allow selective re-optimization of allocation decisions to accommodate new higher-priority missions while minimizing disruption to previous assignments.

As resource assignments are made to a given mission, any necessary auxiliary tasks (for example, positioning/depositioning flights or crew rest periods) are generated and inserted into the mission plan. In the simplest case, all missions are planned and scheduled as round trips. Various missions will be sequenced when necessary to satisfy overall resource capacity constraints (and in some cases rejected as unsupportable). It is also possible to direct the system to consider mission merging possibilities which provides another means for optimizing resource usage. For example, the system might suggest using an aircraft from one mission to support a second mission instead of returning directly back to home station.

Mission scheduling and resource allocation capabilities can be invoked in automated or semi-automated modes. In the latter case, the system generates and compares different options that might be taken. Planners interact with the AMC Allocator through graphical displays, which incorporate mission-oriented, resource-oriented and map-based views of the current set of commitments.

To provide a dynamic scheduler (DS) for use in an execution management context, the AMC Allocator technology has been extended to accept and respond to updated "state of the world" information. The AMC Allocator's GUI was augmented to include an Agenda Panel for displaying, managing, and examining the effects of alerts received from the FMA Executive. Graphical tools were also developed for visualizing the impact of an alert on the existing schedule. The alerts are communicated via OAA to a new message handling module in the DS, which is responsible for computing the effects of an alert on the existing schedule and passing the alerts to the DS UI. This internal architecture is depicted in Figure 2.

While the DS retains the core constraint-based, incremental scheduling architecture of the AMC Allocator, it has been significantly reengineered and extended to incorporate the constraints and resource models that must be taken into account in an execution-management context (for example, airport MOG constraints that dictate how many aircraft can be accommodated simultaneously). Mission itineraries are modeled with much greater fidelity than in the AMC Allocator, introducing new activities such as take-offs, block-ins, preflights, and postflights. In addition, the DS incorporates a more flexible temporal constraint network model than the AMC Allocator. This new flexibility allows for dynamic extension of activities such as crew rests, which in the AMC Allocator were assigned a fixed duration.

Like the AMC Allocator, the DS supports mixed-initiative scheduling, allowing the end user a range of interaction options, from primarily manual with constraint checking, to user selection of system-recommended options for schedule deconfliction, to fully automated rescheduling actions based on predefined user preferences. The DS incorporates all previously developed options for relaxing constraints in circumstances of constraint conflict, such as over-allocating aircraft or aircrews, delaying missions, bumping lower priority missions, or merging multiple missions into a single mission to reclaim capacity. To resolve problems that involve in-process missions, the DS may also add activity delay and itinerary diversion options.

Dynamic Mediator. DM enables the flight manager to make an effective decision by gathering information from other actors quickly. When the flight manager must alter the schedule in response to an unexpected event, time is an important factor because a delayed decision may require the schedule to be altered even more. For example, when faced with a reduction in MOG capacity, the flight manager needs to make a decision that allocates the remaining capacity to the missions that most require it.

The DM module makes two main assumptions:

(1) no single entity possesses all the information relevant to the decision; and
(2) the time allowed for making the decision is limited or a delayed decision is costly.

The originating planners have information relevant to making alterations to the mission schedule that has not been entered into FMA in advance because it is information that is not needed for normal scheduling. For example, for deciding which missions most require the remaining MOG capacity, the cargo contents and the purpose of the mission are often relevant.

Extracting information relevant to decision making is costly because planners must be contacted to extract information. DM automates parts of the process of incrementally extracting only that information that is relevant to the flight manager's decision. The DM lowers the cost of collecting information and computing the correct decision.

Prior to the FMA, the communication was attempted in only the most important decision situations because interpersonal communication was too costly. As a result, the flight manager often makes an educated guess as to the importance of the relevant missions and therefore may make an inappropriate decision based on that guess. The DM module makes communication practical by (1) managing the communication between the flight manager and the planners to focus on relevant information; and (2) storing, organizing, and analyzing the information for the purpose of making a decision. The DM module enables the flight manager to make better decisions during execution, while not precluding the use of personal contact for the most important decisions.

The DM module automates collection of relevant information from planners using queries and replies, implements a search for those queries and replies that minimize the expected communication costs, and enables correct decision making with limited information.

Application Status

We defined a demonstration scenario consisting of several storyboard-level vignettes that illustrate the capabilities of the FMA. The FMA was demonstrated on the vignettes using scripted data feeds that were generated to be as similar to actual data feeds as possible. For instance, one such script uses all 1100 MOG exceptions from the output of an AMC monitoring system. Based on review by subject-matter experts, all of the demonstrated vignettes show useful capabilities beyond what is currently provided by existing AMC flight management software tools.

A brief summary of each vignette follows:

• MOG conflict detected by the FMA Executive and resolved by the execution office and planners with assistance of the DS.
• A single event causes multiple, cascading problems. An airplane breaks on the runway of Airport 1, causing both a wing capacity overallocation problem and a cargo stalled problem. The FMA Executive detects the problems and DS-aided responses must handle multiple problems.
• Multiple events (bad weather and an Instrument Landing System failure) when considered together cause a problem. The FMA detects the problem and suggests responses.
• The FMA monitors system behavior and gives alerts or responds to the situation. For example, FMA might alert when AMC tools that report MOG exceptions and execution-time exceptions are not present or have lost input feeds, or when FMA actors are not present.
• The FMA performs automated responses to a minor problem, controlled by user-established and selected policy.

To give an idea of how the FMA operates, we briefly describe the execution flow of the second vignette above.

Input Event Sequence:
1. The Executive receives a report that the ILS for port P will be offline for a time window [t1, t2] for repairs.
2. The Executive receives a weather exception at P that overlaps with [t1, t2].

- The Executive infers that the airport will be closed for some period due to simultaneous bad weather and no ILS capability. Either event by itself is no problem but together they cause a problem.
- The Executive communicates port closure information to scheduler.
- The Executive queries the Scheduler for affected missions and alerts the Execution user and affected planning offices, customizing alert to each actor.
- The Scheduler automatically computes immediate impact and suggests rescheduling actions:
    - Options include bumping, delaying, over-allocating and re-routing
- The Scheduler computes the “ripple effect” on the downstream schedule.
- The Execution user, possibly collaborating with planning offices using the Dynamic Mediator, selects a schedule fix, after possibly modifying it during interactions with the Scheduler.
- The Execution user and appropriate planning offices are notified of all relevant changes to missions.

The Executive is designed to coexist with and complement the existing flight management software tools currently deployed at AMC. Some existing tools at AMC detect deviations and problems, but they are based on simple rules. Thus, they detect too many false alarms that overwhelm the user with alerts and therefore the user cannot focus on the most important deviations. The FMA improves upon these tools by its VOA computation, which will filter out low-value alerts, and show high-value alerts to those users for whom they have high value. Furthermore, the FMA detects problems that are not detected by existing tools (for example, the closure vignette described above).

Transition tasks. The Executive generally takes inputs in forms that are available in existing AMC tools and databases. The Dynamic Scheduler is already in use at AMC as part of CAMPS. The primary tasks that would be required to transition this technology are as follows:
• The Executive must integrate and interface with any data sources to be monitored.

• The FMA system operates in real time, but must be made more robust with respect to tracking and reasoning about current time (currently use scripted simulated times).
• Design and implementation of an interactive alert/collaboration GUI or integration with existing GUIs must be accomplished.
• Policies must be encoded to implement AMC procedures; additionally, it may be desirable to monitor additional data sources.

Evaluation and Summary

Subject-matter experts determined that the alerts generated and schedule repairs completed using the FMA were correct and valuable in each of the vignettes. The Executive (1) monitors all exceptions from multiple tools, (2) estimates the value of each possible alert, and (3) issues high-value alerts that focus user attention on key problems. Using actor-specific VOA, it effectively filtered and prioritized the alerts generated by existing AMC tools. For example, we ran the Executive and SAAM actors on 1085 actual MOG alerts. The Executive filters all but 242 of the 1085 alerts, of which only one is highest priority, and only eight require immediate attention. The Executive sends the SAAM actor 145 alerts, all of which are lower priority.

Such filtering greatly reduces the amount of information humans must monitor, allowing the humans to concentrate on more important tasks than monitoring large amounts of incoming information. Timely alerts result in faster and better responses to unexpected events. Using the DS to assist with modifications results in more missions being accomplished, more efficient resource usage, fewer constraint violations, and fewer downstream problems. Because the FMA analyzes all inputs against the entire schedule, large, complex schedules can be accurately monitored, and no relevant information is ignored or missed. Finally, our distributed actor architecture ensures that the planners (and other actors) get planner-specific alerts. Thus, planners are kept apprised of the status of their missions and can provide feedback during execution.

The Dynamic Scheduler (DS) provides a range of capabilities for responding effectively and rapidly to exceptional events that have been detected. Upon receipt of an alert from the Executive, the status information contained in the alert is super-imposed over the current existing schedule, and a list of resulting issues (e.g., schedule conflicts) is posted on an agenda panel. As the user selects a given conflict to address, the system invokes graphical displays that indicate the impact of the event. The DS can be directed by the user to generate sets of possible actions for resolving a given schedule conflict (e.g., delay, divert, or coalesce a problematic mission). Alternatively, the DS can be invoked automatically by the Executive (if policy permits) to resolve and/or improve the current schedule. As decisions are made as to which recovery course of action to take, this information is communicated back to the Executive for implementation.

Importantly, policies control system responses; for example, some responses can be made more automated and others more interactive. The coupling of intelligent execution monitoring to dynamic scheduling capabilities introduces several further benefits. Users gain a better understanding of the implications of detected events and prospective responses on other current and planned activities; such implications include projected resource shortfalls, potential mission delays or disruptions, and opportunities for schedule improvement. This coupling also provides rapid generation of alternative recovery actions and more globally rational flight management.

References

Adam Cheyer and David Martin, "The Open Agent Architecture", Journal of Autonomous Agents and Multi-Agent Systems, Kluwer Academic Publishers, Volume 4 (1-2), pages 143-148, 2001.

Laurence A. Kramer and Stephen F. Smith, “Optimizing for Change: Mixed-Initiative Resource Allocation with the AMC Barrel Allocator”, in Proceedings of the 3rd International NASA Workshop on Planning and Scheduling for Space, Houston, TX, October 2002.

Charles L. Ortiz, Jr., Timothy W. Rauenbusch and Eric Hsu (2003). “Dynamic resource-bounded negotiation in non-additive domains.” In Victor Lesser, Charles L. Ortiz and Milind Tambe (Eds.) *Distributed Sensor Networks: A Multiagent Perspective.* In Series: Multiagent Systems, Artificial Societies, and Simulated Organizations, Volume 9. Kluwer Academic Publishers.

Timothy W. Rauenbusch (2004). *Measuring Information Transmission for Team Decision Making.* Ph.D. Thesis, Harvard University.

Stephen F. Smith, Marcel Becker and Laurence A. Kramer, “Continuous Management of Airlift and Tanker Resources: A Constraint-Based Approach”, *Mathematical and Computer Modeling—Special Issue on Defense Transportation: Algorithms, Models and Applications for the 21st Century,* 39 (6-8), 2004.

David E. Wilkins, Thomas J. Lee and Pauline Berry, “Interactive Execution Monitoring of Agent Teams", Journal of Artificial Intelligence Research, volume 18, pages 217-261, March 2003. http://www.jair.org/abstracts/wilkins03a.html.html